

## REMARKS

This Amendment is submitted in response to the non-final Office Action of February 23, 2009. Claims 1-21 are pending. Claim 1 is amended by this response. No new matter is submitted.

### **I        Objections to the Claims**

The Office Action objected to Claim 1. Specifically, the Office Action required “the SetPresentationDescriptor” and “descriptor on the demultiplexer object” to be amended to be “the SetPresentationDescriptor method” and “descriptor of the demultiplexer object” respectively. At least because Claim 1, as amended, recites “the SetPresentationDescriptor method” and “descriptor of the demultiplexer object”, it is requested that these objections be withdrawn.

### **II       Rejections Under 35 U.S.C. §112**

The Office Action rejected Claims 1-21 under 35 U.S.C. §112, second paragraph as being indefinite. Specifically, the Office Action states that it is not clear who is providing and who is utilizing the interface. Further, the Office Action states that it is not clear how the SetPresentationDescriptor method determines not all output associated with an active presentation has been serviced. Applicant respectfully disagrees.

It is respectfully submitted that the Office Action confuses breadth with indefiniteness. A claim is not indefinite merely for being broad. M.P.E.P. §2173.04. It is respectfully submitted it is not necessary to specify “what or whom” is providing or using the interface. Further, it is respectfully submitted that the absence of language in Claim 1 specifying the exact manner in which it is determined, in a particular embodiment, that not all output associated with an active presentation has been serviced does not render the claim language indefinite. An exemplary, non-limiting embodiment in which the computing system executing the computer-executable

Type of Response: AMENDMENT under 37 C.F.R. 1.111

Application Number: 10/635,730

Attorney Docket Number: 302132.01

Filing Date: August 6, 2003

instructions, or any other suitable computing system, provides the interface for the use by the same or a different computing system executing additional or alternative computer-executable instructions and in which the SetPresentationDescriptor method checks a flag, checks a queue or stack, calls a function or takes any other suitable steps to determine that not all output associated with an active presentation has been serviced is clearly within the scope of the relevant language of Claim 1.

For at least the above reasons, it is respectfully requested that these rejections be withdrawn.

### **III Rejections Under 35 U.S.C. §103**

The Office Action rejected Claims 1–6, 12, 15–16 and 18–19 under 35 U.S.C. §103 as being unpatentable in view of “Java™ Media Framework API Guide” (“Java”) in view of U.S. Patent No. 5,455,910 (“Johnson”) in further view of U.S. Patent No. 5,754,774 (“Bittinger”). The Office Action rejected Claims 7–11, 13–14, 17 and 20–21 under 35 U.S.C. §103 as being unpatentable in view of Java in view of Johnson in further view of Bittinger in still further view of U.S. Patent Application Publication No. 2001/0009548 (“Morris”). Applicant respectfully disagrees.

The Office Action acknowledges that Java does not disclose or suggest a SetPresentationDescriptor method to dynamically set an active presentation descriptor of the demultiplexer object to a next pending presentation when an active presentation exists only if all output associated with the active presentation has been serviced, wherein if the SetPresentationDescriptor method is called attempting to set the active presentation descriptor to the next pending presentation when the active presentation exists and not all output associated with the active presentation has been serviced, the SetPresentationDescriptor method indicates that the active presentation descriptor cannot be set to the next pending presentation because not all output associated with the active presentation has been serviced. The Office Action states that this feature is

Type of Response: AMENDMENT under 37 C.F.R. 1.111

Application Number: 10/635,730

Attorney Docket Number: 302132.01

Filing Date: August 6, 2003

disclosed by Johnson. Specifically, the Office Action cites Fig. 10A, elements 105 and 107 and Column 9, lines 42–60 for this feature. However, these sections state:

“The Elaborate Process 91 will now be described with reference to FIGS. 10a-10d. The process is started, step 101, whenever a user executes the New Presentation 41. Next, a flag COMPLETE is set to FALSE, step 103. In step 105, the process gets the first Presentation Description Entry 43 (or the next Presentation Description Entry if the first Presentation Description Entry has already been processed) from the New Presentation 41. Then, the process determines if all of the Presentation Description Entries in the New Presentation have been processed, step 107.

“If the result of step 107 is NO, then the process accesses the Presentation Portion Library entry 47 having the same Portion Name as the Presentation Description Entry that is being currently processed, step 109. The process determines if a Presentation Portion Library entry has been found with a matching Portion Name, step 111. If no such Presentation Portion Library is found, the process reports an error, step 113, and in the preferred embodiment, continues on to get the next Presentation Description Entry, step 105. If the result of step 111 is YES, then in step 115, the memory pointer is set to an allocated memory buffer for the Length. If there is no allocation, that is there is insufficient memory, then the memory pointer is set to 0. In step 117, the method determines if the allocation of memory has failed. If the result of step 117 is NO, that is memory has been allocated, then the process reads the Qualified Source into the memory buffer, step 119. If an Offset and a Length are specified, then that portion of the Qualified Source starting at the Offset is read for the specified Length. The process proceeds to step 121. If the result of step 117 is YES, the process skips step 119 and proceeds to step 121. In step 121, the process determines if the Presentation Portion Library entry function is already loaded. If YES, then the process proceeds to step 135. If NO, then the function is loaded, step 123 and the process determines if there is a load error, step 125. If the result of step 125 is

Type of Response: AMENDMENT under 37 C.F.R. 1.111

Application Number: 10/635,730

Attorney Docket Number: 302132.01

Filing Date: August 6, 2003

YES, there is a load error, then the process reports the error, step 127 and proceeds to deallocate the memory. In step 129, the process determines if the memory pointer=0. If YES, then the process proceeds to step 105 to get the next Presentation Description Entry. If the result of step 129 is NO, then the memory buffer pointed to by the memory pointer is deallocated, step 131, and the process proceeds to step 105.”

It is respectfully submitted that the above sections do not disclose or suggest a SetPresentationDescriptor method to dynamically set an active presentation descriptor of the demultiplexer object to a next pending presentation when an active presentation exists only if all output associated with the active presentation has been serviced, wherein if the SetPresentationDescriptor method is called attempting to set the active presentation descriptor to the next pending presentation when the active presentation exists and not all output associated with the active presentation has been serviced, the SetPresentationDescriptor method indicates that the active presentation descriptor cannot be set to the next pending presentation because not all output associated with the active presentation has been serviced.

The Elaborate Process 91 merely processes data as long as there is data to be processed. The Elaborate Process 91 does not provide any indication that an active presentation descriptor cannot be set to a next pending presentation.

Bittinger discloses flushing a multiplex virtual socket. However, like Java and Johnson, Bittinger does not disclose or suggest suggest a SetPresentationDescriptor method to dynamically set an active presentation descriptor of the demultiplexer object to a next pending presentation when an active presentation exists only if all output associated with the active presentation has been serviced, wherein if the SetPresentationDescriptor method is called attempting to set the active presentation descriptor to the next pending presentation when the active presentation exists and not all output associated with the active presentation has been serviced, the

Type of Response: AMENDMENT under 37 C.F.R. 1.111

Application Number: 10/635,730

Attorney Docket Number: 302132.01

Filing Date: August 6, 2003

SetPresentationDescriptor method indicates that the active presentation descriptor cannot be set to the next pending presentation because not all output associated with the active presentation has been serviced.

For at least the above reasons, it is respectfully submitted that Claim 1 and its dependent claims are patently distinguished from Java in view of Johnson in further view of Bittinger.

Morris discloses a digital video recorder that converts an input data stream having an MPEG-2 Transport Stream format into an output data stream having an MPEG-2 Program Stream (PS) format. The input data stream includes data of at least first and second elementary data streams formed and multiplexed in compliance with a TS decoder model. The first elementary stream is a video stream, while the second stream is an audio stream of lower data rate. The input stream is parsed and the elementary streams are demultiplexed into respective FIFO queues. Each stream is further parsed to obtain and calculate time stamp information which is queued separately at with pointers to the stream data. The schedule and packetisation applied to the elementary streams in the input stream cannot be applied directly in creating the output stream. However, based on parameters and constraints imposed upon the input stream by the MPEG or other specifications, a remultiplexer is able to reschedule and repacketise the elementary data streams into a valid output stream, without the size of buffer that would be required for multiplexing of the elementary streams from scratch.

However, like Java, Johnson and Bittinger, Morris does not disclose or suggest suggest a SetPresentationDescriptor method to dynamically set an active presentation descriptor of the demultiplexer object to a next pending presentation when an active presentation exists only if all output associated with the active presentation has been serviced, wherein if the SetPresentationDescriptor method is called attempting to set the active presentation descriptor to the next pending presentation when the active

Type of Response: AMENDMENT under 37 C.F.R. 1.111

Application Number: 10/635,730

Attorney Docket Number: 302132.01

Filing Date: August 6, 2003

presentation exists and not all output associated with the active presentation has been serviced, the SetPresentationDescriptor method indicates that the active presentation descriptor cannot be set to the next pending presentation because not all output associated with the active presentation has been serviced.

For at least the above reasons, it is respectfully submitted that Claim 1 and its dependent claims are patently distinguished from Java in view of Johnson in further view of Bittinger in even further view of Morris.

#### **IV M.P.E.P. §707.07(j)**

M.P.E.P. §707.07(j) states:

“...If the examiner is satisfied after the search has been completed that patentable subject matter has been disclosed and the record indicates that the applicant intends to claim such subject matter, the examiner may note in the Office action that certain aspects or features of the patentable invention have not been claimed and that if properly claimed such claims may be given favorable consideration...”

Applicants respectfully request that the Examiner make Applicants aware of any subject matter disclosed by the present application which the Examiner believes is patentable. By doing so, the Examiner would help expedite prosecution by enabling Applicants to amend the present claims or draft new claims directed to such subject matter.

#### **CONCLUSION**

Accordingly, in view of the above amendment and remarks it is submitted that the claims are patentably distinct over the prior art and that all the rejections to the claims have been overcome. Reconsideration and reexamination of the above Application is requested. Based on the foregoing, Applicants respectfully requests that the pending claims be allowed, and that a timely Notice of Allowance be issued in this

Type of Response: AMENDMENT under 37 C.F.R. 1.111

Application Number: 10/635,730

Attorney Docket Number: 302132.01

Filing Date: August 6, 2003

case. If the Examiner believes, after this amendment, that the application is not in condition for allowance, the Examiner is requested to call the Applicant's attorney at the telephone number listed below.

Type of Response: AMENDMENT under 37 C.F.R. 1.111

Application Number: 10/635,730

Attorney Docket Number: 302132.01

Filing Date: August 6, 2003

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicants hereby request any necessary extension of time. If there is a fee occasioned by this response, including an extension fee that is not covered by an enclosed check please charge any deficiency to Deposit Account No. 50-0463.

Respectfully submitted,  
Microsoft Corporation

Date: May 26, 2009-----

By: /MacLane C. Key/-----

MacLane C. Key, Reg. No.: 48,250  
Attorney for Applicants  
Direct telephone (703) 647-6566  
Microsoft Corporation  
One Microsoft Way  
Redmond WA 98052-6399

Type of Response: AMENDMENT under 37 C.F.R. 1.111  
Application Number: 10/635,730  
Attorney Docket Number: 302132.01  
Filing Date: August 6, 2003